

FIT 100: Fluency with Information Technology

LAB 8: Intro to JavaScript

Required Reading for Lab 8

- p. 323-331 of FIT Book (part of Chapter 17)
- p. 291-292 of HTML for the World Wide Web
- Forms at w3schools:
http://www.w3schools.com/html/html_forms.asp

References

- Chapters 17 and 18 of HTML for the World Wide Web
- Chapters 1-2 of JavaScript for the World Wide Web

Table of Contents

1.	Setting up your webpage for JavaScript.	1
2.	Hide JavaScript from old browsers using comments	3
3.	Write text onto the page using JavaScript.....	4
4.	Add a Last Modified time/date stamp to your webpage.....	5
5.	Create a Form for your Webpage.....	7
6.	Add Comments Describing Your JavaScript.	9

1. Setting up your webpage for JavaScript.

Before you even begin to write JavaScript you must set up your webpage so that JavaScript will run. In our first HTML lab we learned that you need to put the <html> tag at the beginning of your web page so that the browser knows that it is reading an HTML file. Now that we are about to write JavaScript we must inform the web browser that the script language we will be using is JavaScript.

Look at the source code you find at this location [<http://courses.washington.edu/fit100/au02/labs/lab8.txt>]. Save it as lab8.html on your Desktop. This will be the template you will work from for the next few weeks' labs.

Skill: JavaScript definition:

To alert the browser that the statements that come next are in JavaScript, you will want to use the following basic format:

```
<script type="text/javascript">  
    Your JavaScript Will Go Here.  
</script>
```

You can see that the <script> tag must have a corresponding </script> tag to end the JavaScript.

Where to put your JavaScript

JavaScript code can be written in the <head> of your web page, or in the <body> of your page. Where you put your script depends on what you want to do. Today we will be putting JavaScript into the *body* of our HTML page.

Note: Remember, you can have as many <script> tags as you want so don't feel that you are limited to putting all your code in one place.

- A. Add the beginning and ending script tags to your page just after the second <hr> tag as follows:

```
<hr>  
  <!-- [Exercise 5 will be placed here, so for now just leave the two lines in  
place]→  
<hr>
```

```
  <script type="text/javascript">
```

```
  </script>
```

```
</body>
```

Remember: Code in Bold font is the new code you should add to your page, while regular font indicates pre-existing code.

2. Hide JavaScript from old browsers using comments

Because JavaScript is not always supported by older browsers, we want to “hide” the code by commenting them out. As with CSS, we will insert comment tags around our JavaScript so that if a browser is too old to understand JavaScript it will not display the script on your webpage.

In HTML the characters: `<!--` represent the start of a multi-line comment. When they created JavaScript they made `<!--` represent a single-line comment. This is why JavaScript will ignore this line even though it is within the beginning and end `<script>` tags.

In JavaScript the characters: `//` represent the start of a single-line comment. This means that anything after those two characters (on that line of code) will be ignored by JavaScript. Later you will learn how to do multi-line comments in JavaScript. So, if the browser **can** understand JavaScript, it will ignore this effort to hide the code from the browser. A multi-line comment, in html, ends with the characters: `-->`. Therefore, you need to end the comment with the characters: `-->`, so if the browser **can not** understand JavaScript, it will ignore the script code between the comments to hide the JavaScript.

Note: Adding the texts 'begin comment to hide JavaScript' and 'end comment to hide JavaScript' is a coding convention that allows you or others looking at your code to easily see where the comments begin and end; however, they are not required for the tags to be functional.

Add the comment tags just after the beginning `<script>` tag and just before the end `</script>` tag.

```
<script type="text/javascript">  
  <!-- begin comment to hide JavaScript  
  
  // end comment to hide JavaScript. -->  
</script>
```

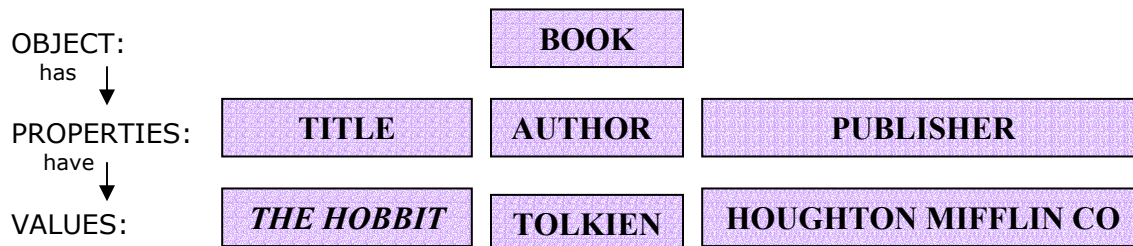
3. Write text onto the page using JavaScript

Using HTML we were able to write text to our webpage using a variety of formatting tags such as the `<p>` tag. Next we will write text on our webpage by writing the JavaScript command **document.write**. Before we do that, let's break down the command itself.

Concept: Understanding Objects and Properties

As you have read, JavaScript is an *Object-oriented programming language*. Objects allow us to organize and manipulate information for components of our program. Most objects in JavaScript have properties. Properties allow us to define a specific instance of an object as having certain qualities. A real world example of an object might be a book. A book might have a title, author, and publisher, just to name a few properties. Each of those items (title, author, publisher) is a property of that book.

Note: The specific title, author and publisher of the book are the **values of the properties**.



In the above example, Book is the object, title, author and publisher are the properties and *The Hobbit*, Tolkien and Houghton Mifflin Co. are the values.

In this lab, *document* is the object we are going to be working with. The document object has properties such as *bgColor* (background color), *fgColor* (foreground or text color) and *lastModified* (the last time the page was modified).

Methods

You will hear and read more about methods during the lectures, but the easy way to think about methods is to look at them as the actions that an object can take. In our example, **write** is an action that the object **document** can take. Using the command **document.write** we will be able to write text to the document. In other words, we will be able to write text to our webpage. The syntax for using `document.write` is:

```
document.write("The text you would like to appear on your page")
```

Note: Every time you write text, the text is considered a **string** and you must put quotation marks around it.

B. In between your beginning and ending `<script>` tags insert the following (bold) code:

```
<script type="text/javascript">
  <!-- begin comment to hide JavaScript

  document.write("This page was last modified on: ")

  // end comment to hide JavaScript. -->
</script>
```

4. Add a Last Modified time/date stamp to your webpage.

You will remember from Project 1 that you were required to put some code on your webpage to display the time your webpage was last modified. Today we are going to add that code to our webpage, but this time we will understand exactly what each part means.

Concept: Variables

Understanding variables is important to programming. We will be revisiting this concept several times over the next few labs and lessons. The best way to think about variables is to think of them as *containers* that store a piece of information. There are two important things to know about using variables:

1. Variables can contain one value. That value might be as short as a single character or as long as a paragraph, however only one value will be stored in a variable at any given time.
2. Before using a variable, you should **declare (or create)** it.

Skill: JavaScript Variable Declaration Syntax

To declare a variable, use the following command:

```
var variablename
```

Note: variable names should not have any spaces.

Note: Declaring a variable announces to your program that you will be using the variable (or container) to store information later in your program.

Skill: Understanding the JavaScript from Project 1

Let's take a look at the code from Project 1:

```
<script type="text/javascript">
  <!-- begin comment to hide JavaScript
  var modified
  modified = document.lastModified
  document.write(modified)
  // end comment to hide JavaScript. -->
</script>
```

<script type="text/javascript">: You should recognize the script tag from Exercise 1

var modified

This declares "modified" as a variable, telling our JavaScript program to create a variable for us to use later in the program.

`modified = document.lastModified` Here we are assigning (or giving) a value to our variable. (We will talk more about assignment later.) You will notice that our *variable* (`modified`) gets the value that is given by referencing a *property* (`lastModified`) of our *object* (`document`).

`document.write(modified)` This line references our *object* (`document`) again and invokes a *method* (`write`) that will write the contents of our *variable* (`modified`) to our webpage.

`</script>` End our script.

C. Insert the following code onto your webpage:

```
<script type="text/javascript">
  <!-- begin comment to hide JavaScript

  document.write("This page was last modified on: ")

  var modified
  modified = document.lastModified
  document.write(modified)

  // end comment to hide JavaScript. -->
</script>
```

5. Create a Form for your Webpage

Forms allow you to create a webpage that have an interactive element. You have all seen forms - from using search engine pages to filling out surveys to buying products online, forms are everywhere. In this exercise we will be laying the groundwork for creating your own interactive webpage. In Project 1 you were asked to have several people rate your webpage. In today's lab we will be creating a survey asking those very same questions.

Forms allow you to produce several input types. A short (and incomplete) list follows:

Text	Button	Checkbox	Radio	Submit
------	--------	----------	-------	--------

These different types of inputs allow your users to enter information into your form. Let's create a few:

D. Between the two `<hr>` tags given to you, place the begin and end Form tags:

`<form>`

`</form>`

E. Add a name attribute to your form. A form is an object just like document is an object and you should always name your objects!

Note: As you learned in your variable assignment, it is always a good idea to name your objects something logical that you can remember and that describes the object you are naming. For example, naming a form "FIT100quiz1" would be good for a FIT100 quiz, but "fred" or "xLw29" would not be as good.

`<form name="YourNameForThisForm">`

`</form>`

F. Add a Text Input box to your form. Like the form, the `<input>` tag has a name attribute. This text box is going to hold the first name of the person who will be filling out your form. Add text before the field so that your user knows what to put in the field.

`<form name="YourNameForThisForm">`

`<p>First Name:`

`<input type="text" name="YourNameForThisTextBox">`

`</form>`

G. Add text fields to hold the following pieces of information:
Note: Remember to name your fields!

- First Name (Done in Step C)
- Last Name
- URL that the person will be rating
- Accuracy Rating
- Authority Rating
- Objectivity Rating
- Currency Rating
- Coverage/Scope Rating
- Accessibility Rating

H. Add a submit button. Remember to name this as well. If you are unsure how to add a submit button, reference your book or one of the websites recommended for the HTML section of the course.

I. Add instructions to your survey so that users will know what to enter in each box. For example, if you want your user to rate things on a scale of 1 to 10, say so. This task will require you to use your knowledge of layout, formatting and HTML to provide a good user interface.

J. As bonus, specify the length of your text boxes to fit your information. That is to say, your text box for the url might be longer than the one needed to rate currency.

6. Add Comments Describing Your JavaScript.

Concept: Comments

There are a number of reasons to put comments in the code that you write. Comments let you describe what your code does, why you made the code choices that you did, and what, exactly, the code does. You might be working on a project with other people and comments will help your team members read your code more easily. Even if you are not working on a team, comments can help you remember what you were thinking when you wrote something. (Picture yourself looking at something you wrote thinking, "what was I thinking??!") In practical terms for this class, comments also help you and the people who are evaluating your work understand the process you went through to get to a specific solution. Even if you can't get something to work, sometimes it is a good idea to leave the code written down within a comment. That way if you realize your error later, you do not have to start all over again.

Skill: HTML commenting:

```
<!-- your comment goes here and can  
span multiple lines -->
```

Skill: JavaScript commenting:

```
// will comment out anything on a single line
```

```
/* will comment out everything between it and the end comment,  
even on multiple lines. This is how to end the comment */
```

A. Write comments for each step of this exercise. Make sure that you use JavaScript commenting when within the `<script>` tags and HTML comments when outside these tags. Explain in detail what each line of code you wrote does, and how it does it. Feel free to explain several lines of code at once if you are comfortable doing so.

Example:

```
<script type="text/javascript">           // start javascript  
<!-- begin comment to hide JavaScript  
  
    var length                           // declares the variable to hold length  
    var width                             // declares the variable to hold width  
    var result                           // declares the variable to hold the result  
    length = 5                            // assigns length the value of 5  
    width = 6                             // assigns width the value of 6  
    result = length * width  
/* result gets the value of length multiplied by width. I will use this variable later in the  
program in multiple places to reference the area of a room. */  
  
    // end comment to hide JavaScript. -->  
</script>
```